HAL
open science

Efficient Low Cost Range-Based Localization Algorithm for Ad-hoc Wireless Sensors Networks

Abdulhalim Dandoush, Mohamed Elgamel

To cite this version:

Abdulhalim Dandoush, Mohamed Elgamel. Efficient Low Cost Range-Based Localization Algorithm for Ad-hoc Wireless Sensors Networks. [Research Report] Inria. 2014, pp.25. hal-01072160v2

HAL Id: hal-01072160

https://inria.hal.science/hal-01072160v2

Submitted on 20 Apr 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Efficient Low Cost Range-Based Localization Algorithm for Ad-hoc Wireless Sensors Networks

*Abdulhalim Dandoush and Mohamed Elgamel

*INRIA Sophia-Antipolis Méditerranée
2004 Route des Lucioles BP93
06902 SOPHIA ANTIPOLIS, France
abdulhalim.dandoush@inria.fr

University of Louisiana at Lafayette
P.O. Box 43004
Lafayette, LA 70504 USA
cshaheen@hotmail.com

Abstract

Building an efficient node localization system in wireless sensor networks is facing several challenges. For example, calculating the square root consumes computational resources and utilizing flooding techniques to broadcast nodes location wastes bandwidth and energy. Reducing computational complexity and communication overhead is essential in order to reduce power consumption, extend the life time of the battery operated nodes, and improve the performance of the limited computational resources of these sensor nodes. In this paper, we revise the mathematical model,the analysis and the simulation experiments of the Trigonometric based Ad-hoc Localization System (TALS), a range-based localization system presented previously. Furthermore, the study is extended, and a new technique to optimize the system is proposed. An analysis and an extensive simulation for the optimized TALS (OTALS) is presented showing its cost, accuracy, and efficiency, thus deducing the impact of its parameters on performance. Hence, the contribution of this work can be summarized as follows: 1) Proposing and employing a novel modified Manhattan distance norm in the TALS localization process. 2) Analyzing and simulating of OTALS showing its computational cost and accuracy and comparing them with other related work. 3) Studying the impacts of different parameters like anchor density, node density, noisy measurements, transmission range, and non-convex network areas. 4) Extending our previous joint work, TALS, to consider base anchors to be located in positions other than the origin and analyzing this work to illustrate the possibility of selecting a wrong quadrant at the first iteration and how this problem is overcome. Through mathematical analysis and intensive simulation, OTALS proved to be iterative, distributed, and computationally simple. It presented superior performance compared to other localization techniques.

Keywords: Wireless Sensor Networks, WSNs, Range-based Localization, Distributed Localization, Anchor-based Localization, Simulation, distance measure, Modified Manhattan, Performance Evaluation

1. Introduction and motivation

The advancement of embedded systems and wireless communication brought forth wireless sensor networks (WSN). These wireless sensor devices are deployed in plethora of applications ranging from civil solutions to military applications. Target localization and real time target tracking are widely adopted applications for wireless sensors networks. Possessing the knowledge

*Corresponding author

of the node's location is a key parameter in these applications and in several routing algorithms [?]. The ad-hoc nature of the sensor devices makes self-positioning an integral part of their applications.

The basic idea used in a node localization system is to measure a physical quantity, e.g. signal strength, that is proportional to distance and use the measured value to estimate the distance to a reference point. Once the distance to at least three reference points is known, the position of the tracked node can be determined.

Thus, the WSN localization process consists typically of three phases (*i*) coordination (e.g. clock synchronization, notification of the need to localize a position) [? ?], (*ii*) measurement of a physical quantity (and the transmission of a signal followed by signal processing), and (*iii*) position estimation [?]. Initializing a localization process requires the coordination of some nodes. One or more nodes then emit a signal, and a property of the signal (e.g. arrival time, phase, signal strength, etc.) is observed by one or more receivers. Next, to determine the location of a node, signal measurements have to be transformed into position estimates by means of a localization algorithm. As mentioned previously, to achieve these phases correctly the existence of cooperating sensor nodes deployed into the environment at known positions a priori is essential. These nodes are called reference points or anchor nodes. The location of an anchor node can be defined through a Global Positioning System (GPS), or manually predefined, or calculated during localization system operations.

Building an efficient node localization system in wireless sensor networks is facing several challenges. For example, reducing communication overhead in the two first phases of the localization process and computational complexity in the third phase is essential to reduce the power consumption, extend the lifetime of battery operated nodes and improve the performance of limited computational resources nodes. Typically, deployments of WSNs exist in harsh environments where nodes are subject to measurement errors due to instrumentation and communication noise. Moreover, WSN nodes are subject to destruction due to environmental effects and dead batteries. To address these challenges and be able to overcome the problems of measurements errors and communication noises of sensor nodes, the redundancy of sensor nodes and information fusion are of great use and are employed in several WSN applications. Node redundancy and information fusion techniques can enhance the quality of the estimates by avoiding outliers and substitute missing information caused by network congestion.

The following is a summary of the challenges faced in building a node localization system

- Flooding: Anchor nodes flood the network with their positions, this wastes energy and bandwidth communication [?].
- Solving a linear system of equations: A system of linear equations is solved using the least square approximations which pose a huge computational overhead [?].
- Finding shortest path: Computing the shortest path between anchor nodes to infer ranging estimates wastes communication bandwidth [?].
- Square root operations: A computational intensive task that does not suit WSN devices.
- High erroneous measurements: That poses a threat on the accuracy of the estimation process.

Therefore, proposing novel simple methods that can improve the localization accuracy with low cost is still a research challenge. In a preliminary joint work in [?], TALS is proposed to address the above challenges. TALS is an anchor based localization system to address the current existing localization systems deficiencies. It utilizes ranging measurements, trigonometric functions and square distance comparisons. Moreover, a data fusion technique is applied to the different redundant estimates of each node to come out with one accurate estimate of the node location. Nodes that their location have been estimated are upgraded to become anchor nodes and aid in the localization algorithm. In this study we introduce a novel distance measurement technique to the TALS localization system, that can be seen as a modified Manhattan (MM) distance measurement technique. MM is used in calculating and comparing two distances as the

localization algorithm under consideration does not need the exact distance calculation for the comparison purpose. It does care about which estimated point is closer to the actual one in order to move to. The optimized TALS (OTALS) localization system is described and its performance is evaluated.

In this paper, we revise the mathematical model,the analysis and the simulation experiments of the Trigonometric based Ad-hoc Localization System (TALS), a range-based localization system presented previously. Furthermore, the study is extended, and a new technique to optimize the system is proposed. An analysis and an extensive simulation for the optimized TALS (OTALS) is presented showing its cost, accuracy, and efficiency, thus deducing the impact of its parameters on performance. Hence, the contribution of this work can be summarized as follows: 1) Proposing and employing a novel modified Manhattan distance norm in the TALS localization process. 2) Analysis and simulation of Optimized TALS showing its computational cost and comparing its accuracy with other related work. 3) Studying the impacts of different parameters like anchor density, node density, noisy measurements, transmission range, and non-convex network areas. 4) Extending our previous joint work, TALS, to consider base anchors to be located in positions other than the origin and analyzing this work to illustrate the possibility of selecting a wrong quadrant at the first iteration and how this problem is overcome. Through mathematical analysis and intensive simulation, OTALS proved to be iterative, distributed, and computationally simple. It presented superior performance compared to other localization techniques.

1.1. Paper Organization

The rest of the paper is organized as follows. In Section 2, the related work in sensor network localization problem is discussed. Section 3 discusses the TALS algorithm in details and presents the Modified Manhattan distance measurement technique to optimize the elimination process; the core process in TALS. This section proposes also a data fusion technique to Optimized TALS (OTALS) results. The analysis of OTALS and the effect of different parameters on the localization process are provided in Section 4. Section 5 describes the simulation environment and presents the results of extensive simulation of OTALS while showing its cost, accuracy and efficiency. The impact of the system parameters on the performance is deduced and the performance is compared with other previous algorithms. Section 6 presents the conclusion and future work.

2. Related Work

Although the literature on localization methods for WSNs is abundant, proposing novel simple methods that can improve the localization accuracy with low cost is still a research challenge and an important issue for several applications in many domains such as commercial, environmental, health, face recognition, image processing and military domains [? ? ? ? ? ? ? ?].

Several localization algorithms exist in literature where many of them depend on variations of Lateration, Trilateration, Multilateration, Angulation or Cellular Proximity. Localization algorithms can be classified into range-free or range-based. Range-based algorithms are more accurate but also more complex and need extra hardware. However, in applications such as target tracking, localization accuracy is important. Range-based methods require at least 3 localized nodes (4 in a 3-D setting) to enable localization of a fourth node with various degrees of quality. The position of wireless nodes can be determined using one of five basic techniques, namely time of arrival (TOA) [?], time difference of arrival (TDOA) [? ? ? ?], angle of arrival (AOA) [?], Hop-count or received signal strength (RSS) [? ? ? ?].

The range-free methods are based on the use of the topology information and connectivity whereas they ignore the use of range measurement techniques [?]. These later methods may use variations of one hop distance victor (DV-HOP) [? ?], an Approximate Point-In-Triangulation test (APIT) [?], Centroid [?], or multidimensional scaling (MDS-MAP) [?]. Those localization schemes are characterized, on one hand, by their simplicity, and ease of implementation, but on the other hand, they have bad localization accuracy [? ?]. J. Lee et al. have proposed in [?] a range-free localization algorithm based on the multidimensional support vector regression

(MSVR). The localization problem is formulated as a multidimensional regression problem, and a new MSVR training method is proposed as a solution of the regression problem. In their method, each sensor node estimates its location using the MSVR model trained with anchor nodes. In the MSVR model,the location of the node is predicted by information related to the proximity of the node to all of the anchor nodes. Two kinds of optimization techniques were proposed to identify the optimal MSVR model. The proposed method was then applied to both isotropic and anisotropic networks and the simulation results justified its good performance.

Anchor-based range-based localization techniques estimate locations usually by solving a set of linear equations using least square approximation or any of its variants. Niculescu and Nath have proposed the APS system in [?] that is an anchor based localization system that utilizes ranging measurements. APS estimates ranges across several hops from the node being localized by finding the shortest path between the anchor nodes and the positioning node. APS utilizes these computed ranges to compute the location of the nodes using a simplified version of the GPS algorithm (see [?] for more details about GPS). The algorithm starts with an initial estimate, and iteratively solves a system of linear equations until the threshold is met. There are several disadvantages for using this technique. APS utilizes flooding techniques to compute the shortest distance from the anchor nodes to each node in the network and to broadcast their positions. This overhead in communications wastes power and time. Furthermore, APS iteratively tries to solve a system of linear equations using least square approximation which presents an overhead in computations. Savvides et al. used the iterative atomic multilateration technique based on TOA in [?] to estimate the location of the nodes. The estimation is calculated by solving a system of linear equation using MMSE [?], when a node has three or more anchor nodes as neighbors. Each node that has computed its estimated location becomes an anchor node and aids in the localization of the rest of the network in an iterative manner. This technique suffers from the complexity residing in solving a system of linear equation which has a complexity of $O(n^3)$ [?].

The TOA and TDOA schemes can achieve high localization accuracy but they require extra hardware and therefore more energy consumption. Among the range-based measurement techniques, the RSS technique is the most common techniques, cheapest and simplest, because it does not require additional hardware. However, it is very susceptible to noise and obstacles, particularly for indoor environment. It requires also more data compared with other methods to achieve higher accuracy as it should consider errors in the measured values, which can be obtained from multi-path propagation, fading effects and reflection. Moreover, extending a RSS-based technique for 3D localization can introduce higher complexity in computational cost and location accuracy [?]. A. Coluccia and F. Ricciato have proposed in [?] a Bayesian formulation of the ranging problem alternative to the common approach of inverting the Path-Loss formula (while considering Received Signal Strength measurements). Numerical results show that the combination of the proposed approaches improves considerably the accuracy of range-based localization that use RSS with only a slight increase of computational complexity.

In [?], the authors have proved first that location estimation variance bounds (Cramer–Rao bounds) decrease as more devices are added to the network. Next, they have shown in a testbed experiment that a sensor location estimation with approximately 1 meter RMS (mean square error) has been demonstrated using TOA measurements. However, despite the reputation of RSS as a coarse means to estimate range, it can nevertheless achieve an accuracy of about 1 m RMS. Fading outliers can still impair the RSS relative location system, implying the need for a robust estimator. The set of these real network measurements are reported in [?] and we have used them for evaluation and comparison purposes with our work. V. H. Dang et al. in [?] have proposed a new distributed algorithm so-called Push-pull position Estimator (PPE). In this algorithm, the differences between measurements and current calculated distances are modeled into forces, dragging the nodes close to their actual positions. Based on very few known-location sensors or beacons, PPE can pervasively estimate the coordinates of many unknown-location sensors. Each unknown-location sensor, with given pair-wise distances, could independently estimate its own position through remarkably uncomplicated calculations. Characteristics of the algorithm are examined through analysis and simulations to demonstrate that it has advantages, in terms of cost-reduction and accuracy improvement, over some previous works. Using a set of real network

measurements, the general performance of this algorithm is compared with our proposed scheme.

The closest work to ours is presented in [?]. In this work, J. Cota-Ruiz et al. have presented a distributed spatially-constrained localization algorithm for WSN (DSCL). Each sensor estimates its position by iteratively solving a set of local spatially-constrained programs. The constraints allow sensors to update their positions simultaneously and collaboratively after each iteration (this part is different from ours) using range and current position estimates to those neighbors within their communications range. Like our algorithm, they introduced a stopping criterion in order to limit the iterations number, and thus, reduce energy consumption with minimal impact on localization accuracy. They identify a square search region at the non-located sensor (as opposed to circular or some other shape) to achieve lower computational complexity on the actual implementation of the algorithm. However, we will show, in Sec. 5, that the performance of our proposed scheme is better than the algorithms introduced in [?] and [?] in terms of accuracy and energy consumption under similar settings.

In [?], C. L. Wang et al. have introduced a recursive least-squares (RLS) optimization process to address localization in WSN, where a recursive-in-time cost function is first defined and then an iterative decentralized algorithm is derived. It is shown that the RLS scheme is equivalent to the incremental subgradient method with an appropriate variable step size for each iteration. To simplify and reduce the high computation cost of RLS, they replaced each gradient value by its "sign" to form a reduced-complexity RLS (RCRLS) scheme. Simulation results show that RCRLS has some performance degradation as compared to RLS, but it is efficient in term of energy consumption. Both RLS and RCRLS require however an important number of iteration to converge to minimum position error. A. Simonetto et al. have proposed in [?] a class of convex relaxations to solve the sensor network localization problem, based on a maximum likelihood (ML) formulation. They have derived a computational efficient edge-based version of this ML convex relaxation class. The solution allows the sensor nodes to solve convex programs locally by communicating only with their close neighbors. They have argued that employing convex relaxations based on a maximum likelihood formulation to message the original nonconvex formulation can offer a powerful handle on computing accurate solutions. Furthermore, they have discussed a distributed implementation of the resulting convex relaxation via the the alternating direction method of multipliers (ADMM).

In our previous joint work, Merhi et al. [?], the 7 point trilateration technique is designed to perform acoustic target localization. Later, this technique has been used in our preliminary joint work in [?] to perform reasonable low cost and accurate node localization. In fact, the 7 point trilateration technique is a range-based anchor-based localization scheme that solves the system of linear equations using geometric representations. As mentioned in Section 1, the objective of [?] was to build a reliable, efficient, and low cost localization system that uses ranging measurements, trigonometric functions, square distance comparisons and a data fusion technique. In this work a novel modified Manhattan (MM) distance measurement technique is employed in TALS instead of the square distance comparison technique in order to provide a more lower cost and simple range-based anchor-based localization system ormance of the system is deeply analyzed and the effects of the different parameters on the efficiency and accuracy are assessed.

Distance measurement techniques (i.e. Euclidean, Cosin and Manhattan) are used in WSN during the estimation phase of the sensor locations [? ? ? ? ? ?]. They are used also in many other fields such as image retrieval [?], Gabor Jets-based Face Authentication (face recognition systems) [?] similarity measurement for motion tracking in a video sequences, estimation of location in WSN , clustering analysis [?], and in the localization process in WiFi AP range [?]. R. Fonseca et al. have proposed in [?] a practical and scalable technique for point-to-point routing in WSN. This method, called Beacon Vector Routing (BVR), assigns coordinates to nodes based on the vector of hop count distances to a small set of beacons, and then defines a distance metric on these coordinates. BVR routes packets greedily, forwarding to the next hop that is the closest (according to this beacon vector distance metric) to the destination. Weighted Manhattan distance functions is used to compute the distance for greedy forwarding. Jie Yu et al. in [?] have first analyzed the relation between distance metric and data distribution. New distance metrics are next derived from harmonic, geometric mean and their generalized forms are presented and

discussed. Those new metrics are tested on several applications in computer vision and it was found that the estimation of similarity can be significantly improved. The evaluation of seven distances for Gabor Jets-Based Face Authentication was performed in [?], where the authors concluded that the performance of a given distance strongly depends on the concrete pre-processing applied to jets and that Manhattan (or city block) distance can outperform cosine, euclidean and other measure distances in some contexts. In [?], X. Wei et al. found that traditional distance functions, e.g., Manhattan and Euclidean, cannot accurately or effectively compute the distance for Routing protocols in Large WSN. Thus, they have presented XLR, a new, flexible and comprehensive framework that realizes important design principles for efficient landmark-based routing while using a parametric-p-norm distance function (PPN) to compute distances. The study [?] presents a new method for off-line signature identification. Fourier Descriptor and Chain Codes features are used in this method for represent Signature image. Signature identification classified into two different problems: to recognition and verification. Different distance measurements to evaluate the results of the recognition process are used including Cosin, Euclidean and Manhattan distances, where Manhattan distance gave the best results. The authors in [?] studied various distance measures and their effect on different clustering Algorithms (for Image Retrieval or Data Mining). They presented a comparison between them based on application domain, efficiency, benefits and drawbacks. In [?], the authors develop a WiFi fingerprint-based localization system using ZigBee radio. They first collects fingerprints (RSS measurements of WiFi APs) from different pre-known locations, and then infers the location based on the observed RSS measurements, through finding the closest match of the collected measures. The authors apply and evaluate three different methods to measure the distance based on the Euclidean distance and the Manhattan distance. They conclude that using Manhattan gives a higher accuracy compared to the Euclidean distance in addition to the fact that Manhattan requires less computation resources and time. However, we claim that our Modified Manhattan formula is novel and specific for the addressed research domain; localization problem in WSN.

3. TALS Algorithm

3.1. Overview

In this section, we describe anchor based localization algorithm that utilizes range measurements, which is called TALS [?]. Some formulas that should be used in the calculations will be corrected also. In particular, the nodes were reported in [?] with their absolute positions and the base anchor node was considered to be located at the origin point (0, 0).

Initial Anchors have their pre-known accurate positions. They are either equipped with GPS or manually predetermined their exact positions. Each node is assumed to be capable of generating a distance estimate to any of its one-hop neighbors. However, nodes are only required to estimate ranges between its location and the anchor nodes. Typically, ranging estimates are performed by one of the four basic techniques. Initially, the algorithm starts by anchor nodes transmitting their own locations to their one hop neighbors. Each node that receives 3 or more anchor locations will start the localization process. Each node selects an anchor node at random to act as a base. The node's estimated position lies on a circle centered at the base anchor node's location with a radius $\tilde{d}$ which is the estimated range to the node being localized as shown in Fig. 1(a). The challenge arises in determining which point on this circle is the estimated position and in which quadrant this point exist. Thus, an elimination process is proposed and employed as in the next sub-section.

3.2. TALS: basic elimination process

First, the basic simple, low cost and efficient TALS elimination process is described. Second, an additional enhancement and optimization in term of computation cost and complexity while conserving the accuracy level is introduced.

First of all, the circle is divided into points as shown in Fig. 1(b), where each point could be a potential estimate and we will refer to it as an estimate point. The division is done uniformly

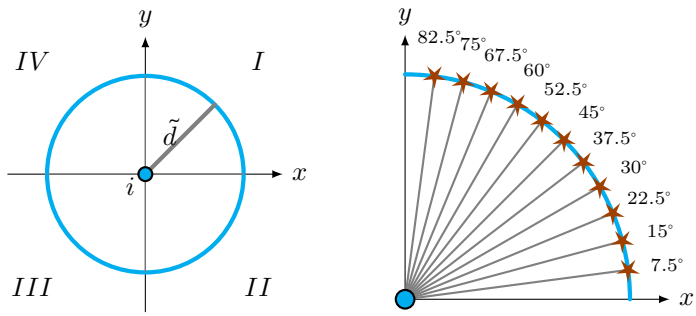


Figure 1: TALS generation of estimate points

across all quadrants of the circle where only the division of quadrant I is shown in Fig. 1(b). The angle degree for which the circle is divided is a design parameter that will be discussed in Sec. 4.2.

To choose which point on this circle is the best estimate and eliminate the others candidate points, additional anchor nodes aid in the localization process. We define $\mathbb{A}$ as the set of these additional anchor nodes. The elimination or the selection is done by comparing the squared estimated range of the node being localized to the squared distance from the additional anchor nodes to the points on the circle through multiple iterations. The elimination process starts by choosing which quadrant the estimate lies on. As shown in Fig. 2(a), consider the base anchor node i and two other anchor nodes $l \in \mathbb{A} = \{j, k\}$. The algorithm starts initially by selecting four points which lie on the circle of node i at 45° from the x-axis as shown in Fig. 2(b,c).

The coordinates of the first estimate point in quadrant I can be calculated using the following equations:

$$\tilde{x}_{45I} = x_i + \tilde{d}_i \; cos\alpha, \; where \; \alpha = 45 \tag{1}$$

$$\tilde{y}_{45I} = y_i + \tilde{d}_i \; sin\alpha, \; where \; \alpha = 45; \tag{2}$$

The three other estimate points that lie on the other three quadrants can be populated using the trigonometric properties:

$$\tilde{x}_{45II} = \tilde{x}_{45I} \quad , \quad \tilde{y}_{45II} = -\tilde{y}_{45I} \tag{3}$$

$$\tilde{x}_{45III} = -\tilde{x}_{45I} \quad , \quad \tilde{y}_{45III} = -\tilde{y}_{45I} \tag{4}$$

$$\tilde{x}_{45IV} = -\tilde{x}_{45I} \quad , \quad \tilde{y}_{45IV} = \tilde{y}_{45I} \tag{5}$$

But as each sensor to be localized will take a basic anchor as the reference for its measurements, we need to adapt the previous equations as below:

$$\tilde{x}_{45II} = \tilde{x}_{45I} \tag{6}$$

$$\tilde{x}_{45III} = \tilde{x}_{45IV} = x_i - \tilde{d}_i \; cos\alpha \tag{7}$$

$$\tilde{y}_{45IV} = \tilde{y}_{45I} \tag{8}$$

$$\tilde{y}_{45II} = \tilde{y}_{45III} = y_i - \tilde{d}_i \; sin\alpha \tag{9}$$

The node will compute (in the basic TALS) the squared distance from anchor j and k to the 4 points using the following equations (see Fig. 2(b,c)):

$$\tilde{d}^2_{j45m} = (x_j - x_{45m})^2 + (y_j - y_{45m})^2 \tag{10}$$

$$\tilde{d}^2_{k45m} = (x_k - x_{45m})^2 + (y_k - y_{45m})^2 \tag{11}$$

for m=I, II, III and IV

The above values are compared against the estimated ranges $\tilde{d}_j^2$ and $\tilde{d}_k^2$ respectively by taking the absolute difference as follow:

$$\tau_m = |\tilde{d}_k^2 - \tilde{d}_{k45m}^2| + |\tilde{d}_j^2 - \tilde{d}_{j45m}^2| \tag{12}$$

for m=I, II, III and IV, the name of a quadrant.

The point that has the lowest residue τ resulting from taking the sum of absolute differences is the point that is the nearest to the node and hence the quadrant is selected as shown in Fig. 2(d). This is due to the fact that for two-dimensional localization, we need usually three anchor nodes (three range measurements from known positions). Each range can be represented as the radius of a circle, with the anchor node situated at the center. Without measurement noise, the three circles would intersect at exactly one point, the location of the target node. Note that we do not solve a linear system of equations and we do not use square root functions.

The reason why the squared distance is used in (10), (11) and (12) lies in the fact that square root functions are expensive in terms of computational complexity. In this way, these functions are eliminated and are substituted for them by squared functions that improves the performance without reducing the accuracy. As shown in Fig. 3, the second iteration of the elimination process starts by selecting two new estimate points on the circumference of the selected quadrant, e.g. quadrant I in this example, located at $\pm 15^\circ$ to/from the first selected point that was lying on 45° in our example. That is, the two estimate points lie on the circle of node i at 30° and 60° from the x-axis as shown in Fig. 3. The coordinates of these two estimate points are calculated using equations (1) and (2) by substituting for α by 30° and 60°. The squared distance to these two points are then computed using the following equations.

$$\tilde{d}_{a30}^2 = (x_a - x_{30})^2 + (y_a - y_{30})^2, \quad \forall a \in \mathbb{A} \tag{13}$$

$$\tilde{d}_{a60}^2 = (x_a - x_{60})^2 + (y_a - y_{60})^2, \quad \forall a \in \mathbb{A} \tag{14}$$

Subscript a in equations (13) and (14) refers to the anchor node aiding in the localization process. The computed values are again compared against the estimated range where the point with the lowest residual τ is selected. If at any time the residual τ is smaller or equal to the threshold ξ, the elimination process stops and the point with the lowest residual is selected as the final estimate. The process continues iteratively in a binary-search like manner until the threshold condition is satisfied. For an example, if point 60° is selected and the threshold condition is not met, the third elimination phase selects the two points (52.5° and 67.5°) such that the comparison will be between them as shown in Fig. 3(d,e,f). Therefore, in any new iteration (i), the selection of the two estimate points (p_{i1} and p_{i2}) is done based on the nearest estimate point computed in the previous iteration (p_{i-1}) using equation (15):

$$\begin{aligned} p_{i1} &= p_{i-1} + \tfrac{15^\circ}{2^{i-2}} \\ p_{i2} &= p_{i-1} - \tfrac{15^\circ}{2^{i-2}} \end{aligned} \tag{15}$$

where $i \geq 2$ stands for the iteration number of the elimination process, and the second term of the right sides represents the degree of divisions so-called $\delta_i = \frac{15^\circ}{2^{i-2}}$ that is the smallest angle with which the iterative division of the circle of the base anchor node is achieved. The algorithm stops when the threshold condition is met or when the divisions reach 0° or 90°. The effect of the parameter δ will be discussed in Sec. 4.2.

3.3. Optimization for the elimination process using a novel modified Manhattan distance

Several distance measurement techniques can be applied to measure the distance between two points (i.e. estimate and anchor nodes). One of the most popular is Minkowski distance that is expressed in Eq. (16). Given two vectors u and v of the same length $N \geq 1$, Minkowski distance is computed as follows:

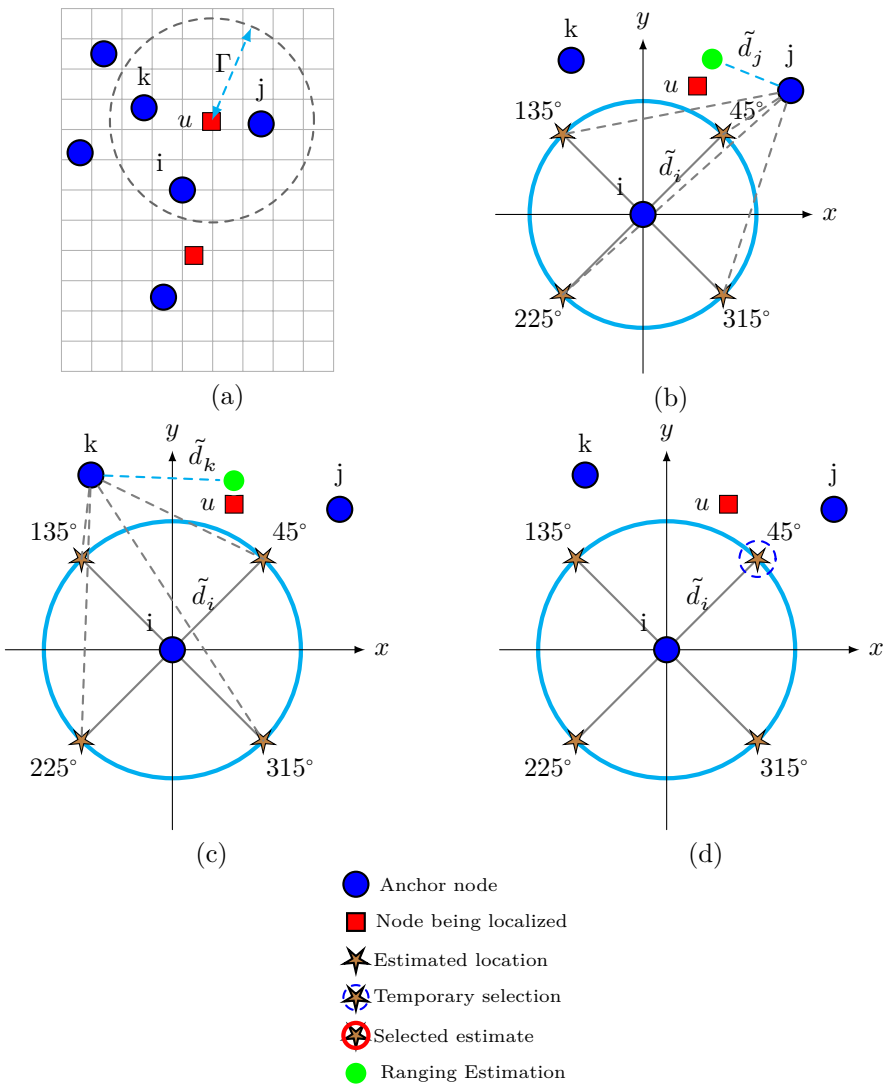


Figure 2: TALS generation of estimate points.

$$dist(u,v) = Minkowski_p(u,v) = \sqrt[p]{\sum_{i=1}^{N} d(u_i,v_i)^p} = \sqrt[p]{\sum_{i=1}^{N} |u_i - v_i|^p} \tag{16}$$

where p is the order of Minkowski distance. Manhattan or city block distance is a special case of Minkowski when $p = 1$ and when p=2, the Minkowski distance is so-called Euclidean distance. In fact, the Euclidean distance gives an accurate measure because it represents the length of the line segment connecting two different points. However, the cost is relatively high because of the use of the root square and the sum of squares as we will see in Sec. 5.9.

In this section, a novel distance measurement technique is proposed to compute the residue τ that indicates the nearest estimate point in each iteration to the real position of the node being localized. The novel Modified Manhattan Distance technique (MM) replaces the Euclidean distance employed in TALS through all the iterations of the elimination process. Our aim is to conserve the accuracy while reducing the cost and the complexity as explained in Sec. 3.4 and Sec. 5.

In fact, τ in (12) was based on equations (10), (11), (13), and (14) that use the Pythagorean theorem; it requires the square function. The new comparison will be given through the following

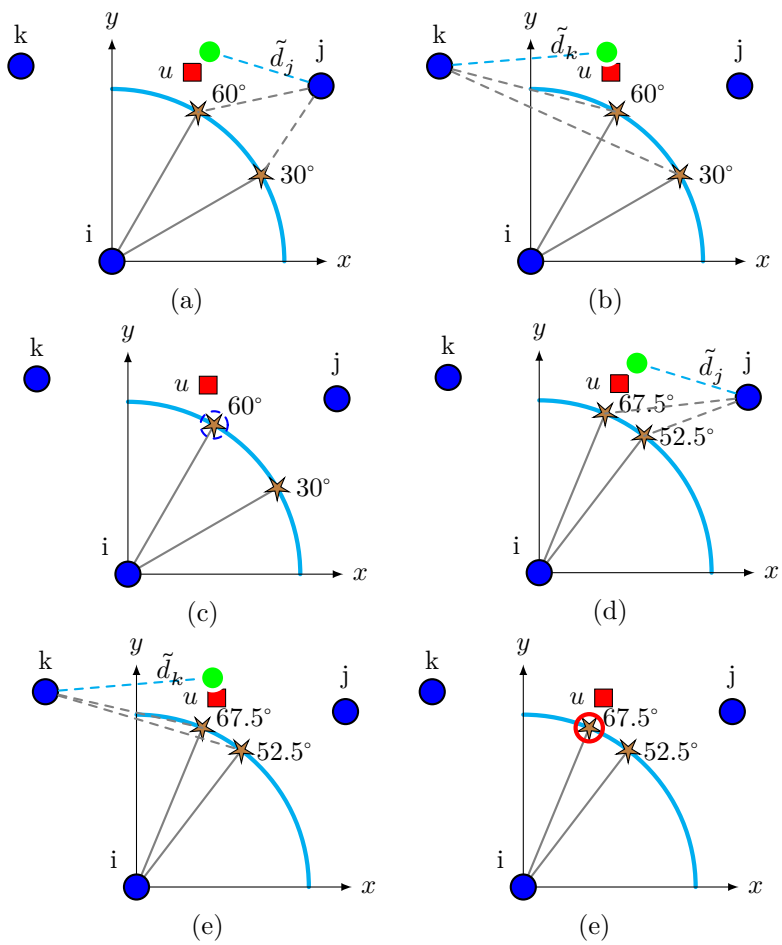


Figure 3: Example of the second iteration phase in TALS elimination process.

equation

$$\tau_p = \sum_{j \in \mathbb{A}} |\tilde{d}_j - \tilde{d}_{jp}| \quad , \tag{17}$$

where p represents an estimate point in a given iteration, $\tilde{d}_j$ denotes the estimated range of the node being localized by the anchor $j \in \mathbb{A}$, and $\tilde{d}_{jp}$ denotes the estimated distance between anchor node j and an estimate point p. This later distance will be computed using a modified Manhattan instead of the squared Euclidean distance, that was utilized in the basic elimination process, as shown in Eq. (18).

$$\tilde{d}_{jp} = \tilde{d}_{X_{jp}} + \tilde{d}_{Y_{jp}} + \theta |\tilde{d}_{X_{jp}} - \tilde{d}_{Y_{jp}}| \tag{18}$$

where $\tilde{d}_{X_{jp}} = |x_j - x_p|$ and $\tilde{d}_{Y_{jp}} = |y_j - y_p|$ and θ is a real constant. The first and the second terms in (18) represent Manhattan distance and the third term represents a corrective factor that will ensure a higher accuracy level of the comparison. Manhattan sometimes gives wrong estimations as shown in the following examples where our modified Manhattan gives more accurate comparison results. The next sub-sections prove that a very good value θ can be set to, in terms of accuracy and complexity, 0.5.

In order to highlight the advantage of the proposed MM over Manhattan, in term of accuracy, three different scenarios of the 3 nodes location (one anchor $j \in \mathbb{A}$ and two estimate points p_1 and p_2) are considered and the Manhattan, MM, and Euclidean distances are applied to determine whether P_1 or P_2 is closer to j. In the three scenarios, we assume the Cartesian coordinates of the

Table 1: Scenarios settings and results

Scenario	Cartesian coordinates	$\tilde{d}_{jp_1}$, $\tilde{d}_{jp_2}$			Comparison result
	p_1, p_2	Euclidean	Manhattan	MM	True(T)/False(F)
1	(2,6), (4,4)	$\sqrt[2]{40}$, $\sqrt[2]{32}$	8, 8	10, 8	Manh(F), MM(T)
2	(5,7), (1,10)	$\sqrt[2]{74}$, $\sqrt[2]{101}$	12, 12	13, 15.5	Manh(F), MM(T)
3	(2,5.2915), (4,4)	$\sqrt[2]{32}$, $\sqrt[2]{32}$	7.2915, 8	8.9373, 8	Manh(F), MM(F)

anchor j to be (0,0) and that the value of $\theta = 0.5$ in (18) for the MM distance. Table 1 resumes the three scenarios settings and results.

In the first scenario, we consider the Cartesian coordinates of the estimate points to be (2,6) and (4,4) respectively. The estimate point p_2 is closer than p_1 to the anchor node j because the Euclidean distances between j from one side and p_1 and p_2 from the other side are $\sqrt[2]{4+36}$ and $\sqrt[2]{16+16}$ respectively. Using the Manhattan formula, these distances are equivalent because dist(j,p_1) = 2+6 = 8 and dist(j,p_2) = 4+4 = 8. Therefore, Manhattan gives a wrong comparison result in this scenario. However, our formula of MM (18) gives the right result because $\tilde{d}_{jp_1} = 2+6+\frac{4}{2} = 10$ and $\tilde{d}_{jp_2} = 4+4+\frac{0}{2} = 8$. Let us presume a second scenario where p_1 and p_2 are (5,7) and (1,10). Here, the estimate point p_1 is closer than p_2 to the anchor node j because the Euclidean distances are $\sqrt[2]{74}$ and $\sqrt[2]{101}$ respectively. Using the Manhattan formula, we find the opposite result; another wrong estimate. Now, using MM we find that $\tilde{d}_{jp_1} = 5+7+\frac{2}{2} = 13$ and $\tilde{d}_{jp_2} = 1+10+\frac{9}{2} = 15.5$, and thus it gives an accurate comparison result.

We show now that MM gives in some particular scenarios wrong results as Manhattan. The frequency of such a wrong result is very low as demonstrated in the next subsection. We consider p_1=(2,5.2915) and p_2=(4,4). The Euclidean distances between j from one side and the estimate points are calculated as $\sqrt[2]{4+28}$ and $\sqrt[2]{16+16}$ respectively. Using the Manhattan formula, the distances are dist(j,p_1) = 2+5.2915 = 7.2915 and dist(j,p_2) = 4+4 = 8. Similar to Manhattan in such a particular scenario, MM cannot capture the true result because $\tilde{d}_{jp_1} = 2+5.2915+\frac{3.2915}{2} = 8.9373$ and $\tilde{d}_{jp_2} = 4+4+\frac{0}{2} = 8$ and thus Manhattan (resp. MM) decides that p_1 (resp. p_2) is closer than p_2 (resp. p_1) to j.

In the next subsection, we first evaluate the comparison accuracy of our proposed MM formula as function of θ. We quantify the error frequency made when using MM instead of the Euclidean distance. Second, we compare through simulating the relative error made by MM with respect to that made by Manhattan.

Fortunately , the elimination process does not depend on a single comparison result from one anchor node. In Equ. (17), τ_p depends on the comparison results of at least two anchor nodes. We demonstrate through simulation in Sec. 5, which addresses the validation and the performance evaluation of the overall OTALS, that the optimized elimination process ensures a very good accuracy with less complexity than the basic process and other localization algorithms found in the literature (to the best of our knowledge).

3.4. Accuracy Evaluation of the Modified Manhattan technique

We will now analytically evaluate the accuracy of our enhanced version of the elimination process using intensive simulation through Matlab that will help to better understand the impact of the wrong result on the localization process.

3.4.1. Analytic analysis

Consider two points $A = (a, b)$ and $B = (c, d)$. We know that A is closer (resp. farther) from the origin $(0, 0)$ than B if $a^2 + b^2 < c^2 + d^2$ (resp. $a^2 + b^2 > c^2 + d^2$).

Question: when do we make a mistake if we replace the above criterion by the criterion "A is closer (resp. farther) from the origin than B if $a + b + \theta|a - b| < c + d + \theta|c - d|$ (resp. $a + b + \theta|a - b| > c + d + \theta|c - d|$) with $\theta \geq 0$?

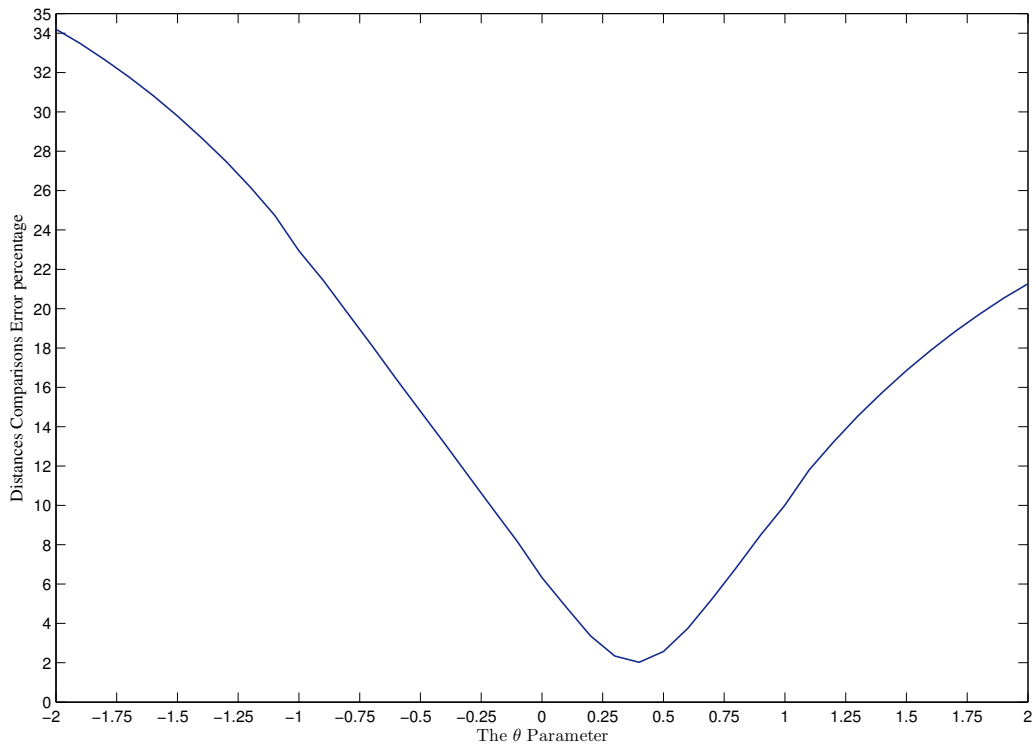


Figure 4: Analysis of Accuracy of Modified Manhattan as function of θ.

Define $X := a^2 + b^2 - (c^2 + d^2)$ and $Y := a + b - (c + d)$.

It is easily seen that $X < 0$, or equivalently A is closer than B from the origin, if and only if

$$Y < \frac{2(ab - cd)}{a + b + c + d}. \tag{19}$$

We therefore need to compare the criterion in (19) to the criterion

$$Y < \theta|c - d| - \theta|a - b|. \tag{20}$$

With (19) and (20) the answer to the question above is easy.

If $\theta|c - d| - \theta|a - b| < \frac{2(ab-cd)}{a+b+c+d}$ then using criterion (20) will give the wrong answer for all points A and B such that $\theta|c - d| - \theta|a - b| < Y < \frac{2(ab-cd)}{a+b+c+d}$. Similarly, if $\theta|c - d| - \theta|a - b| > \frac{2(ab-cd)}{a+b+c+d}$ then using criterion (20) will give the wrong answer for all points A and B such that $\frac{2(ab-cd)}{a+b+c+d} < Y < \theta|c - d| - \theta|a - b|$.

In summary, criterion (20) will give the wrong answer for all points A and B such that

$$\min\left(\theta(|c - d| - |a - b|), \frac{2(ab - cd)}{a + b + c + d}\right) < Y < \max\left(\theta(|c - d| - |a - b|), \frac{2(ab - cd)}{a + b + c + d}\right).$$

Let us now quantify the error that is made when using the criterion (20) instead of the Euclidean distance. To do that, consider a square of size 1 and let us calculate the ratio $R(\theta)$ of pair of points in this square for which criterion (20) will return a wrong answer.

We have (with $A = (x, y)$, $B = (u, v)$)

$$R(\theta) = \int_{x=0}^{1}\int_{y=0}^{1}\int_{u=0}^{1}\int_{v=0}^{1} dx\,dy\,du\,dv\,1\left(\min\left(\theta(|u-v|-|x-y|), \frac{2(xy-uv)}{x+y+u+v}\right) < x+y-(u+v) < \max\left(\theta(|u-v|-|x-y|), \frac{2(xy-uv)}{x+y+u+v}\right)\right).$$

Fig. 4 shows the ratio $R(\theta)$ for $\theta = -2 : 0.1 : 2$. We notice that the best value of θ is equal to 0.4 (It corresponds to 2.0232% of error i.e. the minimum). However, from the practical point of view, setting the value of θ to 0.5 (the second best value that corresponds to 2.574% of error) allows the engineers as shown in the next subsection to replace the complex computations (roots, square, multiplication , division, etc) by a simple shift process (dividing over two in a binary representation) with a minor additional comparison error. Note that the MM becomes Manhattan when $\theta = 0$. From Fig. 4 we see that Manhattan gives 6.325% of the wrong results.

3.4.2. Choice of θ's value

We introduce in this subsection the reason of setting θ to 0.5 instead of 0.4 since equations (17) and (18) do not give enough evidence of saving additional computational complexity with this setting. Multiplication and division operations require many instruction cycles to be performed. In the absence of a hardware multiplier, like a low profile sensor node, multiplication and division can be performed with shift and add operations. Shift-add-multiplication is the simplest way to perform multiplication (see [? ?]). It is derived for binary (n-bits x n-bits) integer which can also be used for single precision floating point number with some minor changes. The number of intermediate addition operations for shift-and-add multiplication method is equal to the number of 1's in the multiplier number. Real numbers are represented by different ways, but the IEEE floating point representation is the most used one. The single precision floating point representation is defined by the IEEE 754 and is represented by 32 bits. These bits are divided into three bit fields: sign bit (1-bit), biased exponent (8-bits), and mantissa (or fraction) (23-bits) as shown in Fig 5. The positive and negative normalized numbers closest to zero are $\pm 2^{-126} \approx \pm 1.17949x10^{-38}$. The finite positive and finite negative numbers furthest from zero are: $\pm(1-2-24)x2^{128} \approx \pm 3.40282x10^{38}$. The sign is stored in bit 32. The exponent can be computed from bits 24-31 by subtracting 127. The mantissa (also known as significant or fraction) is stored in bits 1-23.

Figure 5: Number representation

Figure 6 illustrates the representation of the values 0.5 and 0.4 using the IEEE 754 floating point representation. It is clear from the representation that multiplying by 0.5 only needs a one shift operation compared to multiplying by 0.4 which needs a larger number of shift operations and add operations and, as a result, a larger power consumption. Actually, there are 12 ones in the representation of the mantissa field for the value 0.4, which means 12 add and shift operations are needed, if a simple multiplication algorithm is employed.

3.4.3. Evaluation of MM through Simulation

We now evaluate the accuracy of our enhanced version of the elimination process using intensive simulation through Matlab while considering $\theta = 0.5$. We assume a square area of dimensions 31x31 measurement unit, e.g. m. We take 10218313 random possible locations of two estimate points around an anchor node. The total numbers of the wrong estimations using Manhattan and Modified Manhattan are 623320 and 194704 respectively. The relative errors for both techniques are 6.1% and 1.91% respectively. To understand the reason of the wrong estimations, we first

	Sign	Exponent	Mantissa
Value	+1	2^{-2}	1.600000023841858
Encoded as	0	125	5033165
Binary	0	01111101	10011001100110011001101
Binary	00111110110011001100110011001101		
Decimal	0.4		

	Sign	Exponent	Mantissa
Value	+1	2^{-1}	1.0
Encoded as	0	126	0
Binary	0	01111110	00000000000000000000000
Binary	00111111000000000000000000000000		
Decimal	0.5		

Figure 6: Value 0.5 vs 0.4 representation

maintain two arrays of length equal to the number of wrong estimations using the Manhattan method and the Modified Manhattan method respectively. The value of each element in both arrays correspond to the absolute difference between the straight-line distances of the two estimate points to the anchor node. Then, we draw the histogram of those two arrays that shows the error frequency of the estimation results in both Manhattan and our modified Manhattan techniques with respect to differences between the straight-line distances of the estimate points as demonstrated in Fig. 7. We notice from the figure that the high error frequency occurs when the difference between the straight-line distances of the estimate points to the anchor node is equal to zero, and then to one and two (the estimate points are at the same distance or close to be at the same distance from the anchor node). In fact, with very high probability, this happen in the last iteration phases of the elimination process where the estimate nodes become so close to the right position, and then even if a wrong estimation is made by MM, a very close position is already found. However, and this is a very important remark, when both estimate points are from one distance of one anchor node, they will be with high probability at different straight-line distances from the second anchor nodes that participate in the elimination process. Therefor, the error frequency of the elimination process that depends on the comparison results from at least two Anchor nodes as shown in Eq. (17) will be highly reduced as we will see in Sec. 5. We conclude from the analysis and simulation results that the modified Manhattan is very efficient for hardware of embedded system environment of WSN. This conclusion will be justified and confirmed through accuracy comparison with other previous works and through complexity analysis in Section Sec. 5. It saves more than the 50% of the basic operations number (i.e. binary add operation) used in the calculation of the distance between two nodes similar to Manhattan. In addition, it is more accurate than the Manhattan distance and less complex than the Euclidean technique and other methods appeared in the literature, that require basically solving a linear system of equations via least square or its variant or require a high relative number of iterations for the distributed formulation in previous works.

3.5. Selection of base anchor

In this section, we show how a base anchor is selected. Initially, the ordinary node u that is not localized yet finds n surroundings anchors within its transmission range Γ as in Fig. 2(a). Only if ($n > 2$), node u starts the localization process by selecting one of its surroundings anchors as its base. The highest priority of neighbor anchors to be a base will be given to the fixed anchors. Now, we define two methods to select a base anchor and describe each one in details. The two

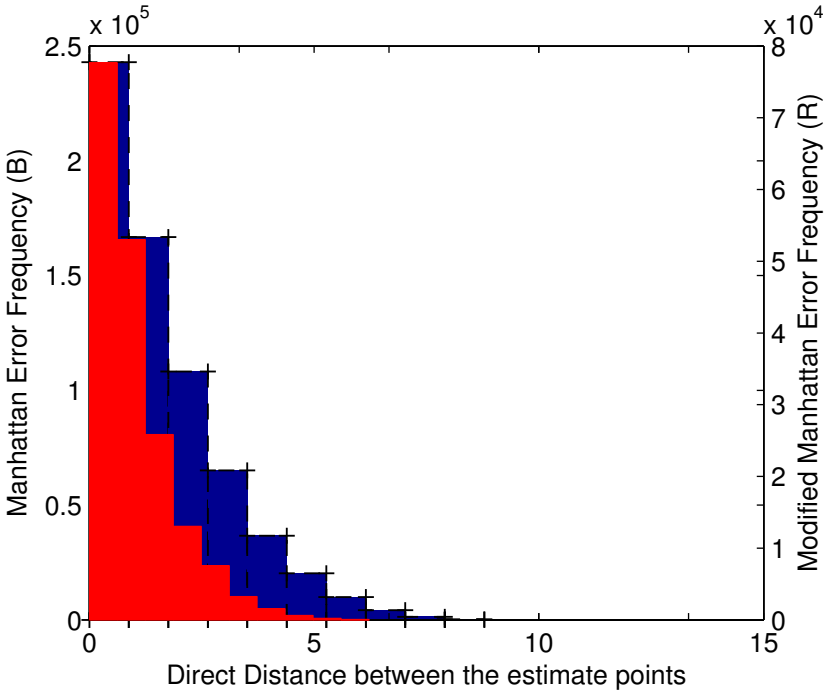


Figure 7: Comparison between Modified Manhattan and Manhattan distances.

methods differ in the computational complexity and storage space.

Method 1: Selecting one base anchor randomly from all neighbor anchors. Fig. 2 shows an example in which node u chooses anchor node i to be its base anchor and then starts to compare the estimated positions on the base anchor circle circumference with radius $\tilde{d}_i$ to the distance from the other anchor nodes j and k. Random selection may be useful for wireless networks with nodes having low computational power. However, it may result in significant errors in selecting the quadrant at the elimination process as we will see in Sec. 4.1.

Method 2: Repeat method one with each anchor, considering one anchor as a base at a time, then selecting the one that leads to the lowest residual. This means that the selection process repeats for all anchor nodes where the next anchor will act as a base and the remaining anchors will aid in the localization process. This way, for each anchor node received by the localizing node, an estimate will be produced. This redundancy will be of a great benefit where fusion techniques can be employed to combine the estimates into an accurate location as shown in Sec. 3.6.

Note that in case the sensor is surrounded by more than three anchors we can consider taking r anchors at a time where $(3 \leq r \leq n)$, where n is the maximum number of anchor nodes in the range of the given sensor. In this case the total number of the combinations is $(\sum_{i=3}^{n} {}^{n}C_i)$. a key issue to be considered is the simplicity of the localization algorithm because it is crucial for implementation and efficiency purposes.

3.6. Data Fusion Technique and Accuracy Enhancement

As shown in Sec. 3.5, TALS repeats the location selection procedure for each anchor node by setting them as base nodes one at a time. Thus, for each anchor node location received, an estimate for the node being localized is calculated. This redundancy enhances the quality of the estimates and eliminates diverging measurements to a great extent. Since TALS computes the locations of the nodes iteratively, weights can be assigned to add confidence to anchor nodes.

The first step in TALS algorithm starts by anchor nodes broadcasting their locations. Each node that receives more than three anchor nodes' location can start the estimation process of its own location. These nodes are upgraded to become virtual anchor node status and broadcast their locations to neighboring nodes and the localization process continues until all nodes are localized. Confidence is given according to two parameters: the number of fixed or virtual anchor nodes and the distance to the node being localized. Higher confidence is given for fixed anchor nodes since they have accurate positions. Thus, given the set of fixed anchors $F_{anchors}$ and the set of virtual anchors $V_{anchors}$, the probabilistic weight is given according to equation (21) where ϕ represents an empty set.

$$w_a(i) = \begin{cases} \frac{0.75}{|F_{anchors}|}, & \text{if}\{\text{i}F_{anchors}\}\&\{V_{anchors} \neq \phi\} \\ \frac{1}{|F_{anchors}|}, & \text{if}\{\text{i}F_{anchors}\}\&\{V_{anchors} = \phi\} \\ \frac{0.25}{|V_{anchors}|}, & \text{if}\{\text{i}V_{anchors}\}\&\{F_{anchors} \neq \phi\} \\ \frac{1}{|V_{anchors}|}, & \text{if}\{\text{i}V_{anchors}\}\&\{F_{anchors} = \phi\} \end{cases} \tag{21}$$

As shown in equation (21), each anchor node belonging to the localizing node's list of anchor nodes is assigned a probabilistic weight w_a according to its status. The value 0.75 in the equation means multiply by 3 and divide by 4, which is 2 shift operations in binary representation and not a real division. Thus, we keep the simplicity of the low computation cost of TALS. In fact, if the anchor node's list contains only fixed or virtual anchor nodes, each anchor node will receive equal weight assignment. On the other hand, if the anchor list contains mixed fixed and virtual anchors, fixed anchor nodes are assigned higher weights than virtual anchors since the latter can have inaccuracies about their estimated positions. Moreover, low estimated ranges from the base anchor node to the node being localized should be assigned high weights since low ranges minimize errors. In this context, given the distance from the base anchor node to the node being localized d_i and the maximum transmission range Γ, the probabilistic weight w_d is assigned according to equation (22). Consequently, each estimate computed from each anchor node is assigned the normalized weight w_{dN} as shown in equation (23).

$$w_d(i) = 1 - \frac{d_i}{\gamma} \tag{22}$$

$$w_{dN}(i) = \frac{w_d(i)}{\sum_{j=1}^{n_anchors} w_d(j)} \tag{23}$$

Since w_{dN} and w_a are independent, their joint probability distribution is given by (24). The final weight assigned for each estimate generated by an anchor node is given by (25). The final estimate is given by equation (26).

$$P(w_{dN}(i), w_a(i)) = w_{dN}(i) \times w_a(i) \tag{24}$$

$$w(i) = \frac{P(w_{dN}(i), w_a(i))}{\sum_{j=1}^{n_anchors} P(w_{dN}(j), w_a(j))} \tag{25}$$

$$(\tilde{x}, \tilde{y}) = \sum_{i=1}^{n_anchors} w(i) \times (\tilde{x}_i, \tilde{y}_i) \tag{26}$$

4. Analysis

In this section, we analyzing the impacts of different parameters of the optimized TALS on the localization process. Additional parameters that are difficult to formulate analytically are studied using simulation.

4.1. Accuracy of Quadrant Selection

As stated before the elimination process starts by choosing which quadrant the estimate lies in. The algorithm starts initially by selecting four points which lie on the circle of node i at 45° from the x-axis. The point that has the lowest residue resulting from taking the sum of absolute differences is the point that is the nearest to the node, hence the quadrant is selected. Next, we show by example that this step may lead to the wrong quadrant when anchors nodes roughly lie on a straight line. This problem is known in the literature as the collinear anchor problem (see [?], [?]). However, by deployment of Method 2 discussed in the previous section or applying a data fusion technique (presented in Sec. 3.6), we can easily overcome this possible wrong estimation.

4.1.1. Example of wrong estimation

Figure 8 gives an example for the output of TALS/OTALS algorithm. In the figure, there are three Anchors $i = (192, 170); k = (154.7, 169); j = (175.2, 170)$ and a sensor node to be localized at $(188.9, 197.7)$. The figure shows an example of the wrong estimation of the Quadrant when the anchor i is selected to be the base anchor in the elimination process. The sensor node actually lies in quadrant II and OTALS wrongly evaluates it to lie in quadrant III. In other words, in the first iteration, instead of selecting the estimate point $e2$, the algorithm selects $e3$ as it is the point that has the lowest residue τ computed from (17). τ_{e3} = (|177-154.7|)+(|153-169|)+0.5*(|22.3-16|) + (|177-175.2|)+(|153-170|)+0.5*(|1.8-17|) = 41.45 + 26.4 = 67.85 is the lowest value, then τ_{e2} = (|177-154.7|)+(|187-169|)+0.5*(|22.3-18|) + (|177-175.2|)+(|187-170|)+0.5*(|1.8-17|) = 42.45 + 26.4 = 68.85 , then τ_{e4} and the last value τ_{e1}.

4.2. Analysis of the effect of Degree of Divisions and Threshold Selection

The degree of divisions δ is defined as the smallest angle with which the iterative division of the circle of the base anchor node is achieved as shown in Eq (15). For example, the case shown in Fig. 1(b) has the smallest angle δ set to 7.5°. The number of estimate points κ is given by $\kappa = \frac{360}{\delta}$. The selection of δ depends on an acceptable estimation error which is an application-based parameter. Finer granularity of δ will yield lower error which comes at the cost of increasing the iterative computations because the number of the elimination phases increase. Therefore, δ is related with the threshold ξ. It was stated in Sec. 3.1 that if at any time the residual τ is smaller or equal to the threshold ξ, the elimination process stops and the point with the lowest residual is selected as the final estimate. Thus, the selection of the threshold ξ depends on two parameters: the degree of division δ and the maximum transmission range Γ. Higher ranges will yield larger radii which translate into bigger pores between the estimated points. The threshold selection is bounded by equation (27) which is given by Theorem 1 that is presented and proofed by Merhi et. al. in [?].

Theorem 1. Given no errors are introduced on the ranging measurements between the localizing node and the anchor node in a one hop scenario, the maximum error is bounded by equation (27):

$$\xi_{max} = \Gamma\sqrt{2(1 - cos(\frac{\delta}{2}))} \tag{27}$$

Where Γ is the maximum transmission range and δ is the degree of division.

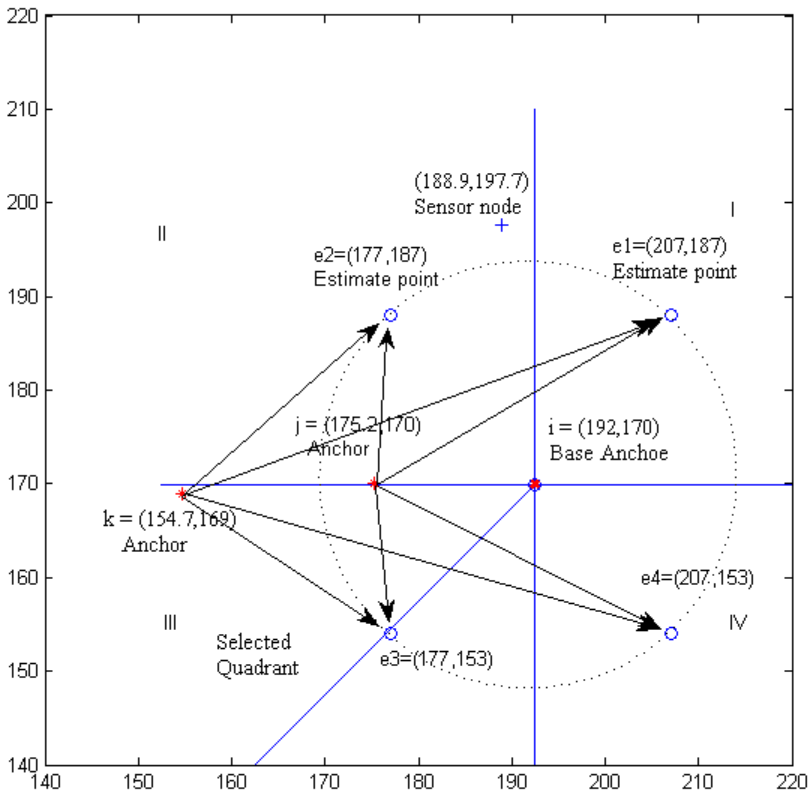


Figure 8: Example of wrong quadrant estimation

4.3. TALS Refinement

There is a trade-off between achieving higher accuracy of the localization algorithm, by reducing the degree of division δ and reducing the computational complexity. In fact, the sine and cosine operations used to determine the estimate points shown in equations (1) and (2) have non negligible computation cost. This section illustrates how computational complexity was decreased by offline pre-calculating the values of the sine and cosine and storing them in the memory of the sensor node. In addition, the trigonometric properties presented in equations (28) and (29) are utilized to further optimize the required number of entries to store the sine and cosine values.

Since the division points on the circle are static and the TALS algorithm tries to select the next point to consider and test for, the sine and cosine values for these points can be stored in a lookup table. However as δ decreases, the number of entries needed to store the sine and cosine values increases, and as a result the size of the look up table increases. Given the the fact that the four quadrants act like mirrors to each other, only 1/4 of the table entries need to be stored. In addition, equations (28) and (29) show that only one entry is needed for the sine or cosine functions. Thus, we need 0.25*360/δ. If $\delta = 7.5$, we need only 12 entries compared to 96 (12 * 4 quadrants * 2 sine and cosine values) entries if the trigonometric properties are not taken into consideration. In fact the optimization can go further using the trigonometric laws and the total lookup table size for each sine and cosine value is given by 45/δ+2, which yield a, 87% saving in the required storage for $\delta = 7.5$.

$$cos(\alpha) = sin(90 - \alpha) \tag{28}$$

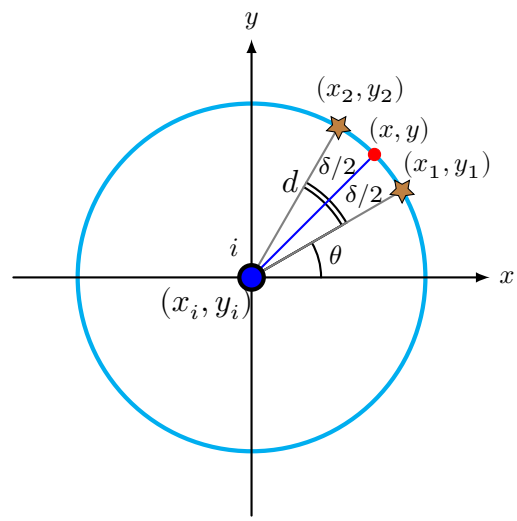


Figure 9: Maximum error in a one-hop scenario.

$$sin(\alpha) = cos(90 - \alpha) \tag{29}$$

5. Validation and Performance Evaluation of the optimized TALS

The performance of the Optimized TALS (OTALS) localization system in term of accuracy is evaluated through a simulation study and a quantitative comparison with other known algorithms for the same localization task (push-pull estimator PPE [?], Least-Squares localization algorithm based on the Levenberg–Marquardt methodology LM as discussed in [?], and Distributed Spatially Constrained Localization-Local DSCL algorithm [?]) using the set of real network measurements reported in [?]. In addition, The computational complexity analysis of the proposed scheme shows the cost efficiency of OTALS. All the simulation scenarios are carried out in MATLAB (The source code is available in [?] for academic use). The next subsection presents the simulation environment, the parameters settings, and the performance metrics.

5.1. Simulation settings

In our simulation, we assumed two different squares deployment areas with sides length 100x100 and 500x500 grid unit. We used a standard uniform pseudo-random generator to distribute the sensors in the deployment area. Note that there are no restrictions for the places where a sensor can be placed, it can be placed at any grid point. We run several simulation experiments using different sensor densities (from 80 up to 250 sensors) in the deployment areas and different percentage of the anchor nodes (from 5% up to 80%). In order to mitigate the randomness, we repeat each experiment 100 times and report the average of the outputs. We stop the simulation when all the nodes having enough neighbor anchors (or virtual anchor) nodes are localized. We consider scenarios with and without noisy range measurements. We assume in some scenarios nodes with homogeneous transmission range of 20, 25 and 35 distance units and in other scenarios with heterogeneous transmission ranges, randomly assigned between 5 and 35 distance units. The localization error is the average of the Euclidean distance between the actual and the estimated positions of sensors. Mathematically, as expressed in (30)

$$Avg.Error = \frac{\sum_{i=1}^{u} \sqrt{(\tilde{x}_i - x_i)^2 + (\tilde{y}_i - y_i)^2}}{u} \tag{30}$$

where u is the number of ordinary nodes, $(\tilde{x}_i, \tilde{y}_i)$ is the estimated node position, and (x_i, y_i) is the actual node position. In all experiments and for any base node i, the degree of division δ is set to 7.5°.

5.2. Validation of OTALS

We will validate the global functionality of OTALS and show how it gives very accurate results in different scenarios. A comparison between OTALS, TALS and other different previously published algorithms is provided and a complexity analysis is introduced. The accuracy and the performance evaluation of the optimized TALS with respect to the different system parameters are addressed.

Figure 10 shows an example of regular WSN topology in a deployment area of 500x500 units and 250 sensors. The number of anchors is 50 nodes (20%). The transmission range is randomly distributed between (10 and 35 unites) and we do not consider noise in the transmission measurements. The figure shows the comparison result between the exact and estimated locations of sensors. We can notice how good the accuracy of OTALS is (96% of the estimated locations overlap the exact ones). For this scenario, we calculated the absolute estimation error vector as the difference between the distance of the exact position of unlocalized nodes to the center and the distance of the estimated position to the center. We reported the mean value of the error and the variance and we depict in Fig. 11 the histogram of this metric. The mean value of the error is 0.3162 measurement unit and the variance is 4.7349 measurement unit. It is clear from the histogram that 246 sensors out of 250 are localized with absolute estimation error less than 2.5 measurement unit.

Figure 12 shows a different scenario where the grid is 100X100 units and the density is 80 sensors with only 10% Anchors (8 Anchors) with transmission range between (5 and 35 unites). The unlocalized nodes that have enough anchor neighbors (3 at least), will become later virtual Anchors and will aid in the localization process of the others. It is clear that OTALS is a very good algorithm even with small number of initial Anchors.

In the following sections, we carefully examine the effects of different parameters on the overall performance.

5.3. Impact of anchor density

The anchor ratio is one of the most important factors affecting the localization accuracy. Thus, we report accuracy and performance results with a varying number of anchor nodes. Since the ratio of the anchor nodes is controlled by the fixed number of sensor nodes deployed in the sensor field, the number of normal nodes decreases if the number of anchor nodes is increased. In this case we are more interested in the performance when the anchor ratio is small because in a practical system the number of anchor nodes will be much lower than the number of unlocalized nodes. Figure 13 shows that different anchor ratios will influence the location errors. One can see that with the same node density at the same deployment area, when the percentage of initial anchors is increased from 20% to 30%, the average localization errors decrease significantly. Our algorithm performs well as it manages to produce less error, even without large numbers of fixed anchors as clearly shown in Fig. 12.

5.4. Impact of node density

Our second goal from the simulation is to determine the effect of node density on the localization success ratio and on the average of the localization error. For this experiment, we gradually increase the number of nodes from 80 to 220 in the same deployment square areas. As the number of nodes grows, the node density increases, and the number of virtual anchors increases too. As depicted in Fig. 13(a), we find that as we increase the node density, the localization success ratio gets higher as expected. The question that has been answered here is: What is a good initial fixed anchor ratio to achieve certain accuracy level or localization error?. We conclude that nodes benefit from more neighbors and redundancy. Also, that the error propagation of ordinary nodes becoming derived anchors and aiding other nodes in the localization process is within an acceptable range.

5.5. Localization under noisy distance measurements

In this simulation, the noisy range measurements are generated randomly and the distance error variation has a maximum of 5% of the actual distance between nodes. As shown in Fig. 13(b), the localization error is plotted against the ranging measurements error. With the increase of ranging errors, the localization accuracy is affected. However, with small noise range measurements, the influence is still negligible in the proposed optimized TALS contradicting many known localization systems.

5.6. Impact of transmission range

Figure 14 shows that OTALS performs better at high transmission ranges where the unlocalized nodes benefiting from more fixed and virtual anchors. The percentage of fixed anchor nodes is kept fixed at 20% and the transmission range in the simulation varies between 10–50 units. Note that for the smallest transmission range available (10 units), the average error is small because many nodes will not be localized (less than three anchor or virtual neighbor nodes are found) and has no error to be calculated. With larger transmission range, more nodes (and potentially all) will be localized and their errors will start to be calculated. However, as we increase the transmission range (above the value that permits most nodes to be localized), average localization error decreases. In addition, with the increase of the network density, the error decreases in a slow rate.

5.7. Localization of non-convex networks

Finally, we evaluate OTALS in non-convex networks. Figure 15 illustrates two instances. The sensors are randomly deployed in a H-shaped, and a C-shaped regions, respectively. We repeat the simulations and the results are consistent to show the applicability and efficiency of OTALS in localization of non-convex networks as shown in Fig. 16.

5.8. Comparison with other previous algorithms

In this subsection, we evaluate the accuracy of OTALS using the set of real network measurements reported in [?]. This data set has been used in other works [?], [?], and [?], making it a good comparison reference. The data set presents a fully connected network (all devices are in range of all other devices) of 40 sensors randomly deployed in an office environment within a 14 by 13 area. Also, this WSN scenario uses four anchors located intentionally in the corners with the goal of increasing their quality [?] and avoiding the collinear anchor problem (see [?], [?]). The data set consists of ToA and RSS measurements. However, we will consider only the case of range-based on ToA measurements. ToA measurement errors are Gaussian with a standard deviation around 1.84 m.

For comparison, we include localization results from the push-pull estimator PPE [?], the Least-Squares localization algorithm based on the Levenberg–Marquardt methodology LM as discussed in [?], the Distributed Spatially Constrained Localization-Local DSCL algorithm [?] and the basic TALS [?]. All these algorithms will be tested using similar procedures. As explained in [? ?], a set of initial positions L^0 is required for the three first algorithms (PPE, LM, and DSCL). Each sensor can compute an initial position using an anchor-based localization scheme similar to the ones reported in [?]. Note that this initial phase is not required for TALS and OTALS, and thus our algorithm economizes energy consumption of this step which is very expensive as explained in [?] and shown in the next subsection. Additional difference between our algorithm and the three algorithms mentioned above is that each sensor must know positions and range measurements to at least three non-collinear anchors and that at the end of each iteration, each sensor transmits its position update to all its one-hope neighbors (this requires a large number of communication messages, and thus energy consumption). In our scheme, a sensor announces its last position estimate only and not the intermediate estimations. Once a position of a sensor becomes known, the node status is upgraded from a non-localized to a virtual anchor node and broadcasts its location to neighboring nodes to help in the localization process. So, if

a sensor receives signals from at least three anchor and/or virtual anchor nodes, it can determine its location.

To test the convergence speed to a minimum estimation error, the first three algorithms were ran for 30 iterations while TALS and OTALS ran for only 10 iterations, and the accuracy of the localization process was assessed using root mean square error (RMSE). The evolution of the RMSE at each iteration is shown in Fig. 17. The figure shows that all the algorithms are converging to a minimum localization error at different rates. The RMSE curve for the PPE scheme has a smooth decay with convergence around 20 to 25 iterations. The LM and DSCL schemes present, with respect to PPE, a fast decay towards a minimum point within 5–7 iterations. OTALS and TALS converges exponentially to a low localization error (related to the value of the angle δ). The first iteration in OTALS corresponds to the identification of the quadrant (eliminate 3 out 4 candidate points). The second iteration corresponds to the value 15° of δ and so on. With $\delta = 15/8°$ in the fifth iteration (see Eq. (15)), the RMSE is 0.9752m and 1.1154m for $\delta = 15/2 = 7.5°$ after the third iteration. Notice that the number of distance comparisons and then computational operation within each iteration in OTALS is less than the others. Because in DSCL, they identify a square search region at the non-located sensor. This region is discretized over a n×n grid. If n=5, a set of 25 candidate solutions for the position is considered. The resolution of the search region is a trade-off between accuracy and time-computation. The performance can be improved by increasing the resolution (e.g. 7×7 candidates) with a corresponding increase in computation. In our case, we have only four candidate points at the first iteration, and later only 2 points for any additional iteration. Thus, if we consider 5 iterations (that corresponds to $\delta_5 = 15/8$) only 9 candidate points for the position are examined with a very good accuracy as shown in Fig. 17. We conclude that OTALS presents superior performance compared to other localization techniques in terms of accuracy and convergence speed. It is worth mentioning that in the previous joint work, Merhi et. al. presented a comparison between the basic TALS System [?], Multilateration [?] and the 7 point trilateration [?] as a function of errors introduced on ranging measurements. TALS performance is superior to all other techniques for the same settings. In the next subsection, we will discuss in details OTALS in terms of complexity and energy saving.

5.9. Computational Complexity Analysis

In this section we compare OTALS to other algorithms in term of complexity. To that end, We provide the number of operations used in the equations of OTALS and the numbers in the selected algorithms to compare with. In particular, we count the number of Additions and Subtractions (jointly referenced as ADDs), Multiplications (MULs), Divisions (DIVs), Square Roots (SQRTs), Shifts (SHIFTs), Comparisons (COMPs), and Absolute Values (ABSs). The complexity of the operations is not the same in terms of the processing requirements they take. In addition, the power consumptions are different for each operation. Multiplication and square root operations have the same asymptotic time complexity using the big O notation [?]. The asymptotic complexity bound does not mean that the operations are equivalent. In fact, it is well known that multiple-precision division and square root operations can be reduced to multiple-precision addition, subtraction, and multiplication by using Newton iteration [? ?]. Moreover, it follows from the Schönhage-Strassen algorithm [?] that square rooting of n-digit numbers can be performed in O (n log n log log n) bit-operations, which is based on the fast Fourier transform. Therefore, in real computation, a square root evaluation is about 5 times more expensive than a multiplication. Also, multiplication tends to be much more expensive than addition. Typically, a multiplication operation takes more cycles than an addition/subtraction operation (about 5 times) [?]. Using HSPICE simulations, the work in [?] compared the power consumption of circuits implemented in CMOS and double pass transistor with asynchronous adiabatic logic (DPTAAL) technologies for different operating frequencies to show that DPTAAL consumes less power. Analyzing the reported results, it is clear that for different logic designs, the power consumption of a multiplier is 15 times the power consumption of an adder for the same word length using the 0.18 μm CMOS. For DPTAAL Technology, the power consumption of a multiplier is calculated to be 5 times the power consumption of an adder for the same word length. These results are the same for both low and high operating frequencies. Similar results showing that a multiplier consumes at least 5

Table 2: TALS Operations Cost

Eq.	Operations	ADD	MUL	SHIFT	ABS	COMP
Eq. 1	$\tilde{x}_{45I}$	1	1	-	-	-
Eq. 2	$\tilde{y}_{45I}$	1	1	-	-	-
Eq. 7	$\tilde{x}_{45III}$, $\tilde{x}_{45IV}$	1	1	-	-	-
Eq. 9	$\tilde{y}_{45III}$, $\tilde{y}_{45IV}$	1	1	-	-	-
Eq. 10 and Eq. 11 for 4 points	$\tilde{d}^2_{j45m}$	3*4*(n-1)	2*4*(n-1)	-	-	-
Eq. 12 for 4 points and selecting the closest point	τ_m	(2n-3)*4	-	-	(n-1)*4	3
Starting Iterations $i = 2, 3, 4$	All rows below repeat 3 times					
Eq. 1 for 2 new points ($\pm\delta_i$°)	$\tilde{x}_{45I}$	1*2	1*2	-	-	-
Eq. 2 for 2 new points	$\tilde{y}_{45I}$	1*2	1*2	-	-	-
Eq. 10 and Eq. 11 for 2 points	$\tilde{d}^2_{j45m}$	3*2*(n-1)	2*2*(n-1)	-	-	-
Eq. 12 for 2 points and selecting a point	τ_m	(2n-3)*2	-	-	(n-1)*2	1
Total When n=4 and for 4 Iterations where $\delta_4 = 3.75°$	-	60+3*32 = 156	28+3*16 = 76	-	12+3*6 = 30	3+1*3 = 6

times the power consumed by an adder for the Ambipolar CNTFET technology can be deducted from [?].

The complexity of the proposed OTALS is compared with the previous work that use the squared Euclidean distance (i.e. TALS) and the Bilateration algorithm BL, and the LM algorithm discussed in [?]. We consider OTALS with $\theta = 0.5$ as explained in Sec. 3.4.2 that replaces the multiplication with a simple shift operation. In addition, we consider that the number of anchor nodes participating in the elimination process of non localized sensors to be n, where n will take the value 4 similar to the work presented in [?]. Even that in TALS and OTALS, having n=3 anchors is enough to localize the sensor nodes. We consider 4 iterations for OTALS/TALS whereas 13 iterations are considered for LM in [?], and the BL is a non-iterative algorithm. We have shown in Sec. 5.8 that OTALS and TALS with 4 iterations give more accurate results than LM with 13 iterations or even more. The other algorithms (PPE and DSCL) use the BL for the initial phase, therefore comparing our work with BL will be enough as DSCL and PPE need more cost. The number of operations required to localize one sensor node using TALS and OTALS are presented in Table 2, and Table 3 respectively. The number 4 appears in calculating (10), (11), (12) in table 2, and (17) and (18) in table 3 is for repeating the operations 4 times for the 4 different points on the 4 quadrants, where m=I, II, III, IV. The DIV and SQRT are 0 in both algorithms and their columns are removed from the 2 tables.

Concerning the number of CPU cycles required for the basic operations as floating point operations, we follow the analysis described in [?]. Table 4 summarizes the considered values in this study. In addition, we consider 1 cycle for each of the SHIFT and ABS operations and 11 cycles, similar to ADD, for the COMP operation.

Table 3: OTALS Operations Cost

Eq.	Operations	ADD	MUL	SHIFT	ABS	COMP
Eq. 1	$\tilde{x}_{45I}$	1	1	-	-	-
Eq. 2	$\tilde{y}_{45I}$	1	1	-	-	-
Eq. 7	$\tilde{x}_{45III}$, $\tilde{x}_{45IV}$	1	1	-	-	-
Eq. 9	$\tilde{y}_{45III}$, $\tilde{y}_{45IV}$	1	1	-	-	-
Eq. 18 for 4 points	$\tilde{d}_{jp}$	5*4*(n-1)	-	1*4*(n-1)	3*4*(n-1)	-
Eq. 17 for 4 points and selecting the closest point	τ_p	(2n-3)*4	-	-	(n-1)*4	3
Starting Iterations $i = 2, 3, 4$	All rows below repeat 3 times					
Eq. 1 for 2 new points ($\pm\delta_i$°)	$\tilde{x}_{45I}$	1*2	1*2	-	-	-
Eq. 2 for 2 new points	$\tilde{y}_{45I}$	1*2	1*2	-	-	-
Eq. 18 for 2 points	$\tilde{d}_{jp}$	5*2*(n-1)	-	1*2*(n-1)	3*2*(n-1)	-
Eq. 17 for 2 points and selecting the closest point	τ_p	(2n-3)*2	-	-	(n-1)*2	1
Total When n=4 and for 4 Iterations where $\delta_4 = 3.75°$	-	80+3*44 = 222	4+3*4 = 16	12+3*6 = 30	48+3*24 = 120	3+1*3 = 6

Table 4: Operations cycle counts [?]

ADD	MUL	DIV	SQRT
11	25	112	119

$$\begin{aligned}\text{Total number of cycles for TALS} \quad = \quad & (11)(156) + \\ & (25)(76) + \\ & (1)(0) \quad + \\ & (1)(30) \quad + \\ & (11)(6) \quad = 3712 \text{ cycles}\end{aligned}$$

$$\begin{aligned}\text{Total number of cycles for OTALS} \quad = \quad & (11)(222) + \\ & (25)(16) + \\ & (1)(30) \quad + \\ & (1)(120) \quad + \\ & (11)(6) \quad = 3058 \text{ cycles}\end{aligned}$$

The number of cycles for the LM and the bilateration BL algorithms discussed in Sec. 6.2 in [?] is 63063 and 14198 respectively. As explained in the previous subsection, these algorithms consume a lot of energy because of the initialization stage. In addition, LM requires high number of iterations to find a solution (e.g. 13 iterations for this calculation) and the bilateration in

Table 5: Comparison of required number of CPU cycles

Algo	LM [?]	BL [?]	TALS [?]	OTALS
cycles	63063	14198	3712	3058

Table 6: Operations power consumption

ADD	MUL	SHIFT	ABS	COMP
5	25	1	1	5

non-iterative algorithm but needs around 2500-3000 cycles to perform the sorting step and requires many SQRT/MUL/DIV/ADD operations. Note also that some recent algorithms such as PPE and DSCL that we discussed in Sec. 5.8 and Sec. 2, use the BL to produce the initial position of the sensors. Thus, the cost of these algorithms is relatively high and we do not need to do the calculation of their cost. Table 5 summarizes the comparison of the number of CPU cycles required for OTALS and the other discussed algorithms.

Comparing the cycles taken by each algorithm is not a reliable measure for the power consumption of the sensor node, which is a main challenge in WSN. Based on the discussion above for [54-58] and [? ?], we can present an estimate of the relative power consumption for each operation as shown in table 6.

$$\begin{aligned}\text{Total power consumption for TALS} \quad = \quad & (5)(156) + \\ & (25)(76) + \\ & (1)(0) \quad + \\ & (1)(30) \quad + \\ & (5)(6) \quad = 2740\end{aligned}$$

$$\begin{aligned}\text{Total power consumption for OTALS} \quad = \quad & (5)(222) + \\ & (25)(16) + \\ & (1)(30) \quad + \\ & (1)(120) \quad + \\ & (5)(6) \quad = 1690\end{aligned}$$

There is approximately 38% saving in power consumption utilizing OTALS compared to using TALS in the localization process. We conclude that OTALS proved to be a promising localization algorithm that one can build on it for providing low complex and accurate localization solution to many WSN applications.

6. CONCLUSION

In this work, a novel low complexity anchor-based range-based localization system based on trigonometric identities and properties that we have recently proposed is intensively analyzed and different optimization techniques are utilized. OTALS is tailored to suit wireless sensor network environment where the contributions made can be summarized as follows:

- Proposing and employing a novel modified Manhattan distance norm in the TALS localization process

- Analyzing and simulating of OTALS showing its computational cost and comparing its accuracy with other related work
- Studying the effects of parameters like anchor density, node density, noisy measurements, transmission range, and non-convex network areas.
- Extending our previous joint work, TALS, to consider base anchors to be located in positions other than the origin and analyzing that work to illustrate the possibility of selecting a wrong quadrant at the first iteration and how this problem is overcome.

  Through the mathematical analysis and intensive simulation, OTALS proved to be iterative, distributed, and computationally simple. It has presented superior performance compared to other localization techniques.

In the future, we plan to adapt and evaluate the OTALS to the 3D-localization problems and to consider the sensors mobility and asses its impact on exchanging data and the performance in VANET.

Acknowledgement

We would like to thank Dr Philippe Nain from INRIA-Sophia Antipolis for his advice and assistance concerning the analysis of Modified Manhattan and other helpful technical discussions. We also appreciate the efforts of anonymous reviewers and the editor of the journal toward improving the paper.

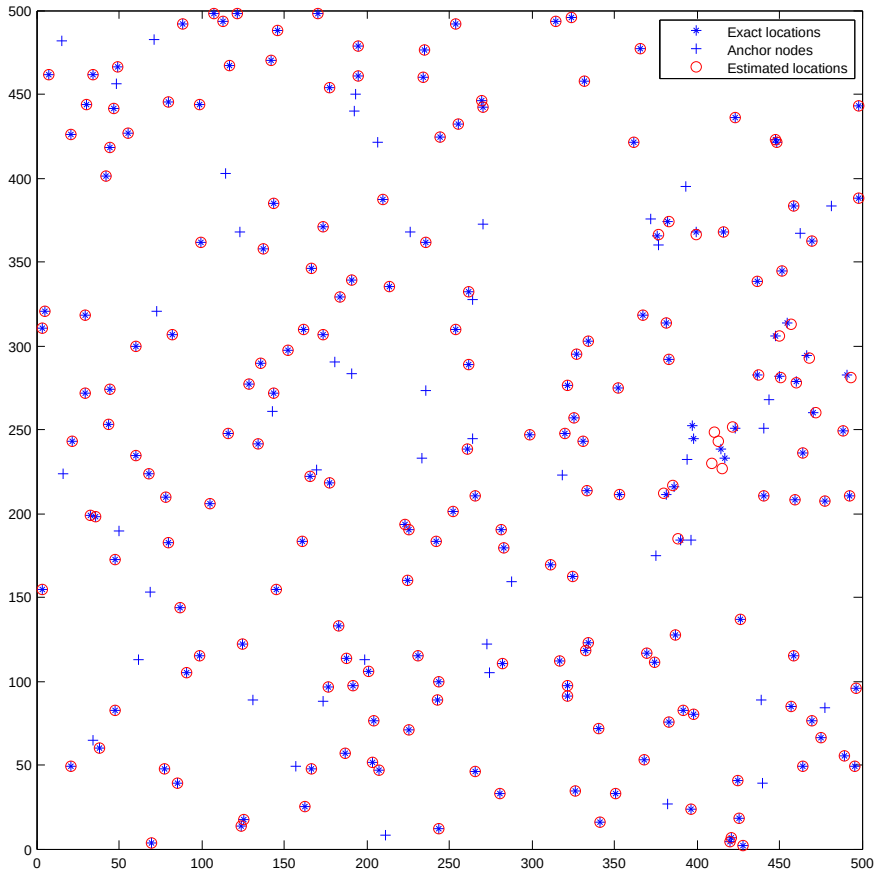


Figure 10: Comparison between exact and estimated sensors location. 500x500 grid, 250 sensors with 20% Anchors (50), Random transmission ranges (10–35)

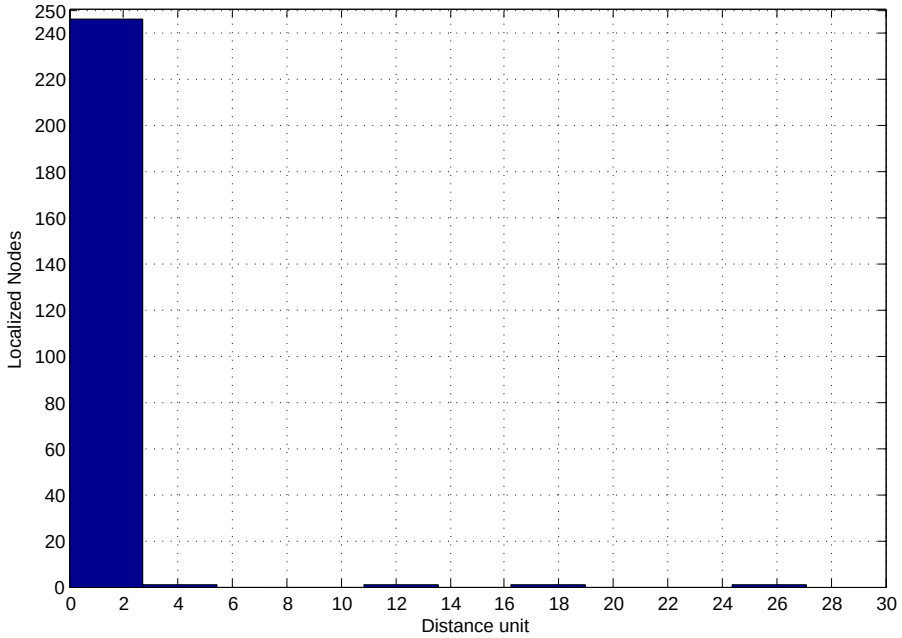


Figure 11: Histogram of the absolute error between the distance of exact and estimated sensors location to the center. 500x500 grid, 250 sensors with 20% Anchors (50), Random transmission ranges (10–35)

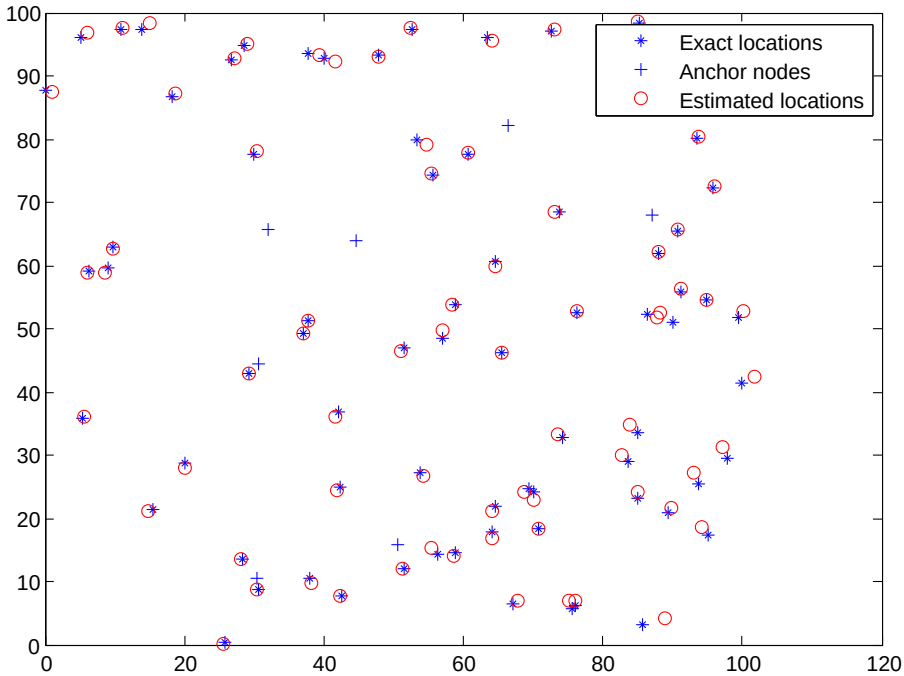
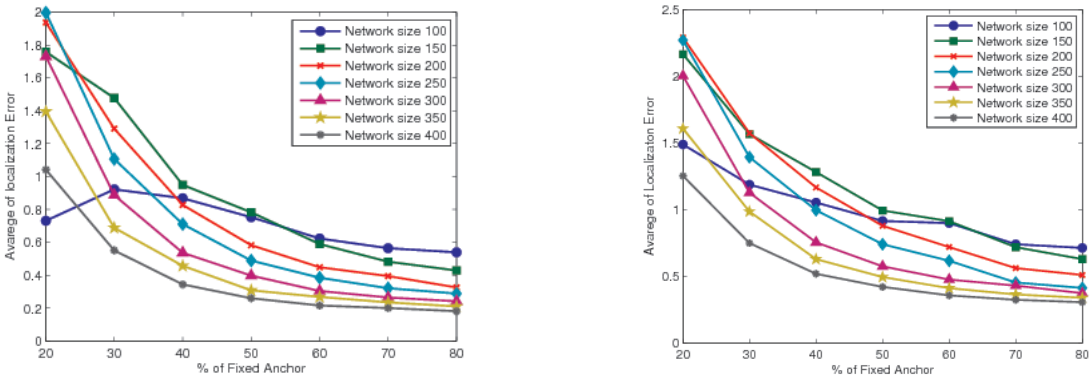


Figure 12: Comparison between exact and estimated sensors location. 100x100 grid, 80 sensors with 10% Anchors (8), Random transmission ranges (5–35)



(a) Average position error without noise interference (b) Average position error with noise interference

Figure 13: Effect of number of Fixed Anchors under different node densities on localization error.

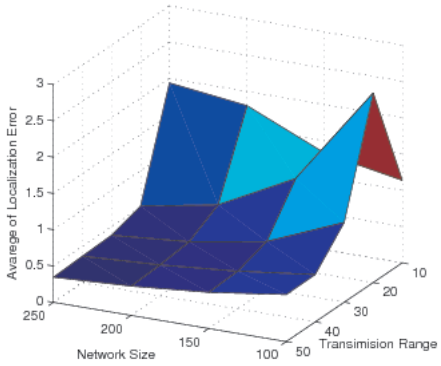
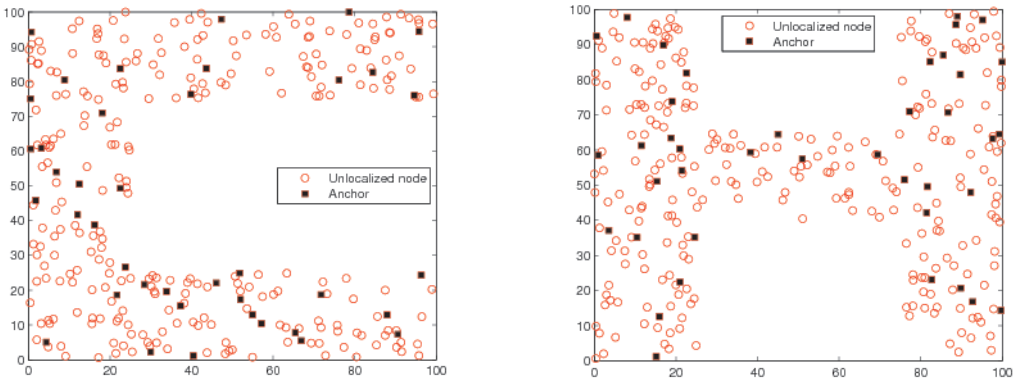
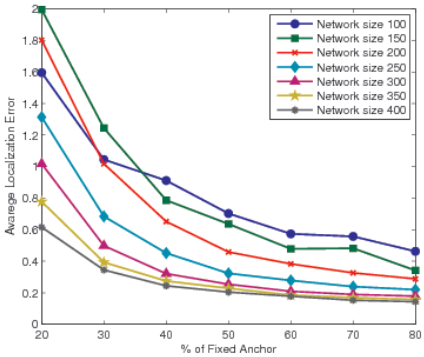


Figure 14: Localization success vs. Network size and Transmission range.

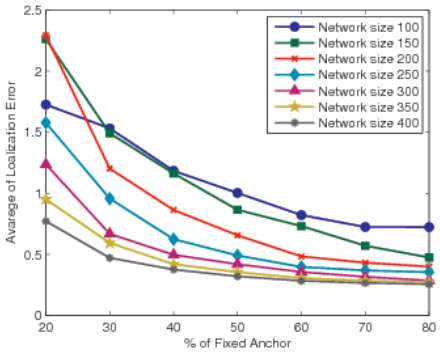


(a) (b)

Figure 15: Irregularly-Shaped non-convex Networks. (a) Example of C-shaped network topology. (b) Example of H-shaped network topology.



(a) C-shaped network topology



(b) H-shaped network topology

Figure 16: Effect of number of Fixed Anchors under different node density on localization error.

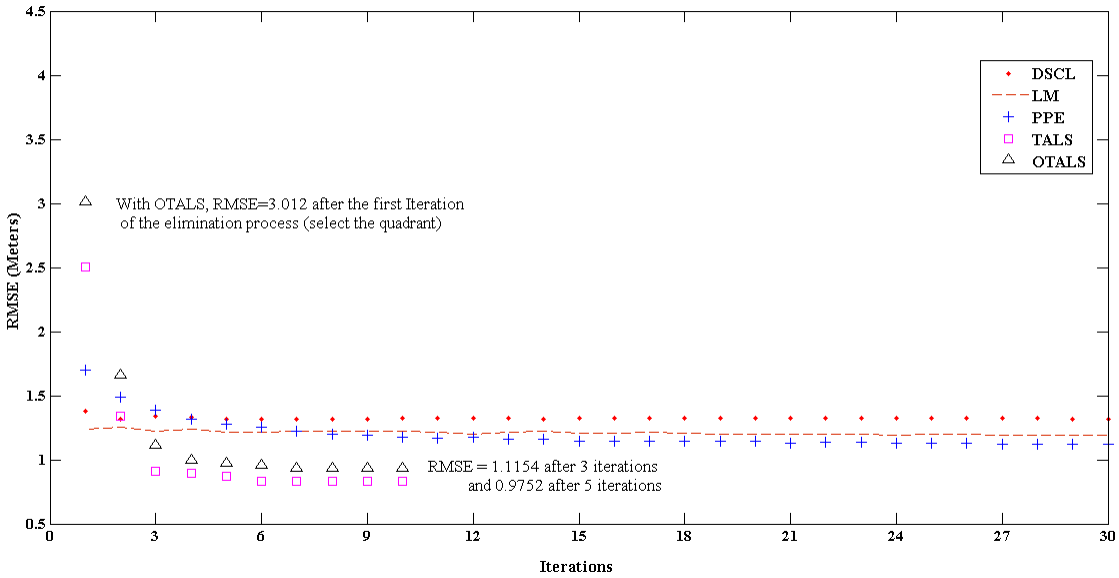


Figure 17: Comparison with previous works